■ ประพนธ์ อัศวกาญจวัฒน์

ActiveX แรงตั้งแต่เกิด : ได้เวลาเรียนรู้และใช้งาน ActiveX Form

ActiveX สร้างสัมพันธ์ระดับทวิภาคี ระหว่างโปรแกรมต่างภาษา ให้สื่อสารภายใต้เน็ตเวิร์กเดียวกันได้ เริ่มจากสร้างฟอร์มใช้งานบนเว็บเบราเซอร์ กับ ActiveX Form

Delphi

ในปัจจุบันเทคโนโลยี COM หรือที่รู้จักกันในนามของ Component Object Module ถือเป็นเทคโนโลยีพื้นฐานที่น่าสนใจมากที่สุด เนื่องจากการพัฒนาที่ต่อเนื่องและก้าวกระโดดทำให้การนำเทคโนโลยีนี้ไปประยุกต์ใช้งานสามารถกระทำได้อย่างเต็มประสิทธิภาพ ดังเช่น เทคโนโลยี ActiveX ซึ่งมีรากฐานที่สร้างด้วย COM เทคโนโลยี เป็นต้น

เมื่อกล่าวถึงเทคโนโลยี ActiveX แล้ว หลาย ๆ ท่านคงสงสัยว่า ActiveX คืออะไร มีความสำคัญอย่างไร ถ้าพิจารณาตามคำจำกัดความที่บริษัทไมโครซอฟท์ให้ไว้แล้ว ActiveX คือเทคโนโลยีที่ช่วยให้โปรแกรมหรือส่วนของโปรแกรมที่พัฒนาด้วยภาษาที่ต่างกันสามารถตอบสนองหรือสื่อสารกับโปรแกรมหรือส่วนของโปรแกรมอื่น ๆ ภายใต้ระบบเน็ตเวิร์กเดียวกันได้ และด้วยเทคโนโลยีในปัจจุบันคงไม่เกินเลยที่จะกล่าวเสริมว่า ActiveX จะเป็นพื้นฐานต่อไปสำหรับการสื่อสารข้ามแพลตฟอร์ม (Platform) ในอนาคตอันใกล้นี้

สำหรับท่านที่มีความสนใจในเทคโนโลยี COM และเนื้อหาของโปรเซส (Process) หรือเธร็ด (Thread) สามารถศึกษาพื้นฐานเพิ่มเติมในส่วนของการใช้งานของ OLE Automation Object ได้จากหนังสือ "Delphi Episode II เทคนิคการพัฒนาโปรแกรมชั้นสูงด้วยเดลไฟ" โดยสามารถสอบถามข้อมูลเพิ่มเติมได้ที่ www.se-ed.com

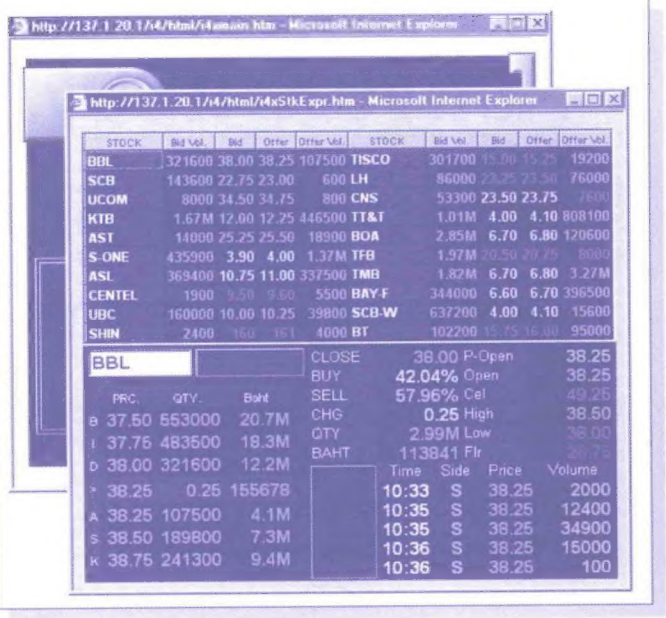
ผู้เขียนตั้งใจจะกล่าวถึงเรื่องเกี่ยวกับการนำเทคโนโลยี ActiveX มาประยุกต์ใช้งานกับโปรแกรม แต่เนื่องจากเนื้อหาของเทคโนโลยีนี้แตกแขนงแบ่งย่อยออกเป็นหลายส่วน ในฉบับนี้ผู้เขียนจึงขออธิบายเฉพาะส่วนของเทคโนโลยี ActiveX Form เท่านั้น สำหรับส่วนอื่น ๆจะขอนำเสนอในโอกาสต่อไป

ActiveX Form

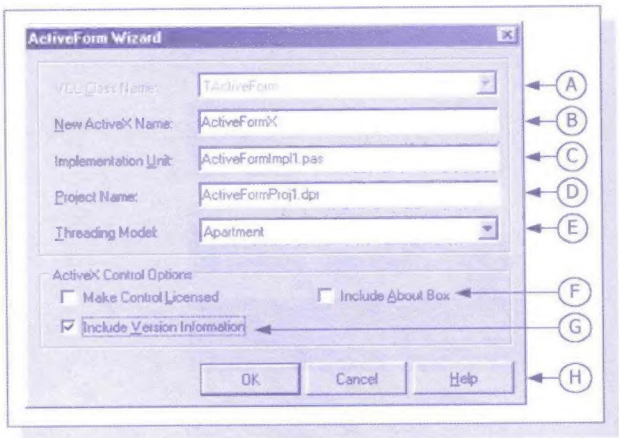
ActiveX Form เป็นหน้าต่างคอนโทรลชนิด ActiveX ที่สืบสกุลจาก TactiveForm โดยมีจุดมุ่งหมายเพื่อสร้างฟอร์มให้สามารถใช้งานบนโปรแกรมเว็บเบราเซอร์ (Web Browser) ที่สนับสนุนการใช้งานเทคโนโลยี COM และ ActiveX เช่น Internet Explorer เวอร์ชัน 4 ขึ้นไป เป็นต้น

การพัฒนา ActiveX Form สามารถใช้แนวความคิดเดียวกันกับการพัฒนาวินโดวส์แอพพลิเคชัน (Windows Application) ทั่วไป กล่าวคือ นักพัฒนาระบบสามารถเพิ่มเติมคอมโปเนนต์มาตรฐานของเดลไฟใน ActiveX Form ได้ตามปกติ เนื่องจากเมื่อมีผู้เรียกใช้ ActiveX Form ผ่านเว็บเบราเซอร์ ActiveX Form จะฝัง (Embeded) ออบเจ็กต์ตัวเองลงไปในพื้นที่แสดงผลของเบราเซอร์ ดังรูปที่ 1

สำหรับข้อแตกต่างระหว่างการใช้งาน TactiveXForm และ Tform คือ TactiveX Form จะอยู่ในรูปแบบ OLE Custom Control หรือไฟล์ OCX ซึ่งจะสามารถทำงานได้บนคอนโทรลคอนเทนเนอร์ (OLE Control Container) เท่านั้น ไม่สามารถรันในลักษณะแอพพลิเคชัน (EXE : Executable File) ทั่วไปได้



รูปที่ 1 ตัวอย่าง ActiveX Form ที่ทำงานบน Internet Explorer



รูปที่ 2 หน้าต่าง ActiveX Form Wizard

ควบคุมการทำงานของ ActiveX Form

ActiveX Form จะใช้เทคโนโลยี COM เป็นอินเตอร์เฟสระหว่าง ActiveX Form กับคอนโทรลคอนเทนเนอร์ที่ติดตั้งอยู่ โดยปกติ ActiveX Form จะใช้อินเตอร์เฟสแบบ In-process server ในการติดต่อกับคอนโทรลคอนเทนเนอร์ ทั้งนี้ขึ้นอยู่กับรูปแบบการทำงานของเธร็ด (Thread Model) ที่เลือกใช้ ดังนี้

Single Thread Model เป็นรูปแบบของ ActiveX Object ที่ไม่ใช้เธร็ดสนับสนุนการทำงาน ทำให้อินเตอร์เฟสที่เกิดขึ้นระหว่างออบเจ็กต์จะถูกควบคุมด้วยเธร็ดหลักของคอนเทนเนอร์

Apartment Thread Model หรือ Single-Threaded Apartmentจะใช้เธร็ดพิเศษสนับสนุนการอินเตอร์เฟสระหว่างออบเจ็กต์ต่าง ๆ ที่ทำงานร่วมกัน โดยออบเจ็กต์ทุกตัวจะถูกควบคุมการทำงานด้วยเธร็ดพิเศษแยกต่างหาก ประโยชน์ที่ได้รับคืออินสแตนซ์ (Instance) และข้อมูลต่าง ๆ (Varies Data) จะได้รับการปกป้องจากโปรแกรมส่วนอื่น และอินเตอร์เฟสระหว่างออบเจ็กต์ต่าง ๆ มีความแน่นอนกว่า เนื่องจากจะถูกควบคุมตามข้อกำหนดของเธร็ด

Free Thread Model หรือ Multi-Threaded Apartment จะใช้เธร็ดพิเศษสนับสนุนการอินเตอร์เฟสระหว่างออบเจ็กต์ต่าง ๆ แบบอิสระต่อกัน กล่าวคือ ออบเจ็กต์แต่ละตัวสามารถมีเธร็ดพิเศษควบคุมการทำงานของตนเอง เป็นผลให้การทำงานของออบเจ็กต์ที่ใช้มีประสิทธิภาพมากขึ้น โดยปกติการใช้งาน Free Thread Model จะนิยมใช้กับการอินเตอร์เฟสจากเครื่องรีโมตอื่นผ่าน DCOM (Distributed Component Object Module)

การสร้าง ActiveX Form

การสร้าง ActiveX Form สามารถทำได้โดยใช้ ActiveX Form Wizard ดังนี้

1. รันโปรแกรมเดลไฟ
2. เลือกเมนู File -> New เพื่อเปิดหน้าต่าง New Items Dialog
3. เลือกแท็บ (Tab) ชื่อ ActiveX
4. เลือกไอคอน ActiveX Form เดลไฟจะสร้างหน้าต่างฟอร์ม Wizard ให้กำหนดรายละเอียด ActiveX Form ใหม่ดังรูปที่ 2 โดย

A. VCL ClassName จะถูกกำหนดให้เป็น TactiveForm

B. New activeX Name สำหรับกำหนดชื่อคลาสให้กับ ActiveX Form ที่สร้างใหม่นี้

C. Implementation Unit สำหรับกำหนดชื่อของยูนิตที่ใช้บันทึกข้อมูลของ ActiveX Form

D. Project Name ใช้สำหรับกำหนดชื่อโปรเจ็กต์ให้กับโปรเจ็กต์ใหม่นี้

E. Threading Model สำหรับกำหนดรูปแบบการทำงานโปรเซสหรือเธร็ดที่คอนโทรลคอนเทนเนอร์ใช้ติดต่อกับ ActiveX Form

F. Include About Box ให้กาเครื่องหมาย ☑ ในกรณีที่ต้องการสร้างหน้าต่าง About Box เพิ่มเติม เดลไฟจะดำเนินการสร้างฟอร์มพิเศษต่างหากเพื่อใช้เป็นหน้าต่าง About Box

G. Include Version Information ให้กาเครื่องหมาย ☑ ในกรณีที่ต้องการให้ ActiveX Form บันทึกข้อมูลส่วน Version Information ซึ่งสนับสนุนการใช้งานในรูปแบบ Auto Update กล่าวคือ เมื่อคอนโทรลคอนเทนเนอร์เรียกใช้ ActiveX Form จะทำการตรวจสอบ Version Information ของ ActiveX Form ที่เคยติดตั้งอยู่ที่เครื่องของคอนโทรลคอนเทนเนอร์กับ ActiveX Form ที่เซิร์ฟเวอร์ก่อน และในกรณีที่พบว่า ActiveX Form ที่เครื่องของคอนโทรลคอนเทนเนอร์ (Clients) มีเวอร์ชันเก่ากว่า ActiveX Form ของเซิร์ฟเวอร์ คอนโทรลคอนเทนเนอร์ก็จะดำเนินการดาวน์โหลดโปรแกรมตัวใหม่ทันที

H. กดปุ่ม [OK] เพื่อเริ่มสร้าง ActiveX Form โปรแกรม Wizard จะดำเนินการสร้างไฟล์ข้อมูลต่าง ๆ สำหรับ ActiveX Form ดังนี้

1. โปรเจ็กต์ ActiveX Form ซึ่งบรรจุโค้ดที่จำเป็นในการสร้างฟอร์มนี้

2. Type Library file (ไฟล์ที่ลงท้ายด้วย TLB) ที่จะบันทึกฟังก์ชันอินเตอร์เฟส พร็อปเพอร์ตี้ กิจกรรม หรือข้อกำหนดต่าง ๆ ที่ ActiveX Form ใช้ติดต่อกับคอนโทรลคอนเทนเนอร์

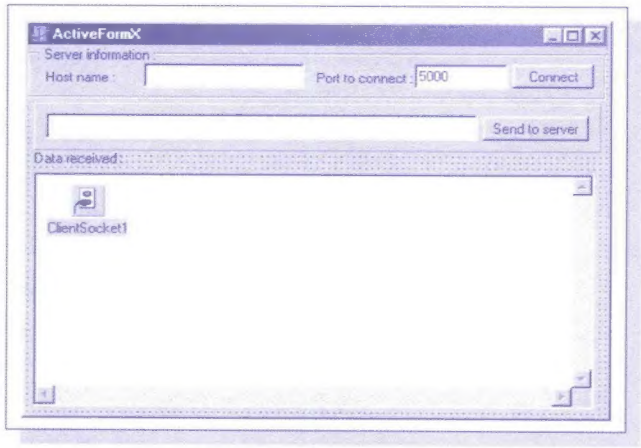
3. ActiveX Implementation Unit เป็นไฟล์หลักสำหรับ Active Form โดยจะบันทึกรายการคอมโปเนนต์ พร็อปเพอร์ตี้ กิจกรรมต่าง ๆ ที่นักพัฒนาระบบสร้างขึ้น และส่วนของโปรแกรมไว้ในยูนิตนี้

ตัวอย่างโปรแกรม

โปรเจ็กต์ ActiveFormProj1 และโปรเจ็กต์ AxServer เป็นโปรแกรมตัวอย่างแสดงการใช้งาน ActiveX Form โดยโปรเจ็กต์ ActiveFormProj1 จะเป็นโปรแกรมที่ทำงานในส่วนของ ActiveX Form บนโปรแกรม Internet Explorer ที่เครื่องลูกข่าย (Clients) และโปรเจ็กต์ AxServer จะเป็นโปรแกรมที่ทำงานในส่วน Service Server ที่ให้บริการกับเครื่องลูกข่าย โดยสื่อสารกันผ่านซ็อกเก็ตด้วยโปรโตคอล TCP/IP

เตรียมการ

การทดสอบการทำงานของโปรแกรมต้องเตรียมคอมพิวเตอร์จำนวน 2 เครื่องที่ติดตั้งอยู่บนระบบเน็ตเวิร์กเดียวกัน ดังต่อไปนี้



รูปที่ 3 หน้าต่าง ActiveX Form : ActiveFormProj1

VCL Class Name	VCL Class Name
New AcitveX Name	ActiveFormX
Implementation Unit	ActiveFormImpl1.pas
Project Name	ActiveFormProj1.dpr
Threading Model	Apartment

รูปที่ 4 ข้อมูลที่ใช้กำหนดใน Wizard

1. เครื่องเซิร์ฟเวอร์จะต้องติดตั้งระบบปฏิบัติการ Windows NT หรือ Windows 2000 ที่ได้ทำการติดตั้ง Internet Information Service (IIS) เวอร์ชัน 4 ขึ้นไป และกำหนด Service Configuration เรียบร้อย เป็นเว็บเซิร์ฟเวอร์ (Web Server) พร้อมที่จะให้บริการระบบอินเทอร์เน็ตหรืออินทราเน็ต

2. เครื่องลูกข่ายที่ได้รับการติดตั้งระบบปฏิบัติการ Windows 98 หรือสูงกว่า ซึ่งสามารถติดต่อไปยังเครื่องเซิร์ฟเวอร์ และขอใช้บริการอินทราเน็ตเป็นที่เรียบร้อย

ActiveFormProj1

ActiveFormProj1.Dpr เป็นโปรเจ็กต์ที่แสดงการทำงานในส่วนของ ActiveX Form บนเว็บเบราเซอร์ของเครื่องลูกข่าย ซึ่งแสดงไว้ในรูปที่ 3 และส่วนของโปรแกรมแสดงอยู่ในตารางที่ 1 (อยู่ตอนท้ายของบทความ) มีขั้นตอนการสร้างดังต่อไปนี้

1. การสร้าง ActiveFormProj1 ให้ดำเนินตามขั้นตอนการสร้าง ActiveX Form ทุกประการ โดยกำหนดรายละเอียดของ Active Form ดังรูปที่ 4

2. เมื่อดำเนินการสร้างเรียบร้อยจะได้หน้าต่าง ActiveFormX และเมื่อไฟล์ ActiveFormImpl1.pas ด้วย Code Editor จะพบว่าเดลไฟได้สร้างส่วนของโปรแกรมที่จำเป็นสำหรับ ActiveX Form เรียบร้อยพร้อมใช้งาน

3. ให้ดำเนินการวางคอมโปเนนต์ที่ต้องการใช้ตามรูปที่ 3

4. สร้างกิจกรรมตอบสนองเหตุการณ์ต่าง ๆ ของคอมโปเนนต์ที่ใช้ตามขั้นตอนการโปรแกรมต่อไป

ยูนิต ActiveFormImpl1.pas มีส่วนของโปรแกรมที่สำคัญดังต่อไปนี้

กิจกรรม TActiveFormX.Button2Click(Sender: TObject); เป็นกิจกรรมสำหรับกำหนดพอร์ตและเครื่องเซิร์ฟเวอร์ที่โปรแกรมจะทำการติดต่อให้กับคอมโปเนนต์ ClientSocket1 โดยกำหนดให้ Port เป็นหมายเลขเดียวกันกับหมายเลขพอร์ตของโปรแกรม axServer ที่เปิดให้บริการอยู่ และ hostName เป็นหมายเลข IP Address ของเครื่องเซิร์ฟเวอร์ด้วยคำสั่ง

```
ClientSocket1.Host:=hostName;
ClientSocket1.Port:=portNo;
ClientSocket1.Open;
```

กิจกรรม TActiveFormX.Button1Click(Sender: TObject); เป็นกิจกรรมสำหรับดำเนินการส่งข้อมูลที่คัดลอกจากคอมโปเนนต์ Edit1 ให้กับเครื่องเซิร์ฟเวอร์โดยผ่าน ClientSocket1 ด้วยคำสั่ง

```
s:=Edit1.text; p:=@s[1];
clientSocket1.Socket.SendBuf( p^, length( s ) );
```

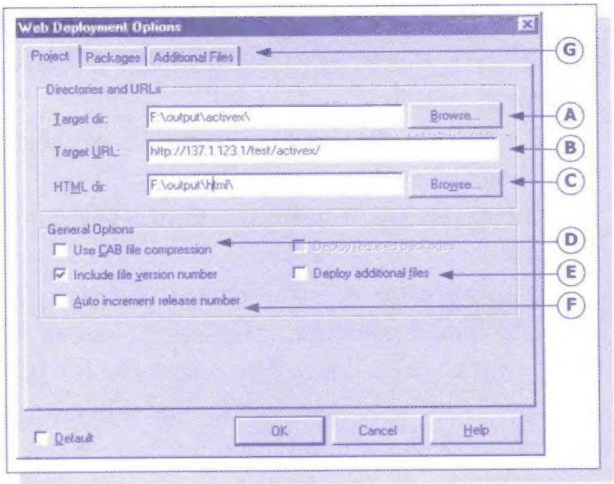
กิจกรรม TActiveFormX.ClientSocket1Read (Sender: TObject; Socket: TCustomWinSocket); เป็นกิจกรรมตอบสนองการรับข้อมูลจากเครื่องเซิร์ฟเวอร์ผ่านคอมโปเนนต์ ClientSocket โดยจะนำข้อมูลที่รับได้แสดงไว้ในคอมโปเนนต์ Memo ด้วยคำสั่ง

```
size:=socket.ReceiveBuf( p^, sizeof( shortString ) );
s[0]:=chr(size); Echo( s );
```

กำหนดพารามิเตอร์และคอมไพล์โปรแกรม

การคอมไพล์โปรแกรมที่สร้างในรูปแบบ ActiveX Form จะแตกต่างกับโปรแกรมทั่วไปที่สามารถเลือกเมนูโปรเจ็กต์ แล้วเลือกเมนูย่อย [Compile] หรือเลือกเมนู [Run] เพื่อรันโปรแกรมโดยตรง แต่สำหรับ ActiveX Form เนื่องจากเป็นฟอร์มที่ออกแบบให้ทำงานอยู่บนคอนโทรลคอนเทนเนอร์เช่นโปรแกรมเว็บเบราเซอร์ Internet Explorer ซึ่งนอกเหนือจากการคอมไพล์โปรแกรมด้วยวิธีปกติเพื่อสร้างไฟล์ OLE Custom Control (OCX) แล้วยังต้องทำการสร้างเว็บเพจควบคุม (Web page) เพื่อใช้เป็นเพจสำหรับการลงทะเบียนเรียกใช้ ActiveX Form รวมทั้งภายในเว็บเพจยังระบุขนาดของฟอร์มและวิธีการเรียกใช้ ActiveX Form

การกำหนดพารามิเตอร์เพื่อเตรียมส่วน Web Deploy สามารถทำได้โดยการเปิดหน้าต่าง Web Deploy Options ด้วยการเลือกเมนู Project —> Web Deploy Options.. เดลไฟจะแสดงหน้าต่างดังรูปที่ 5 เพื่อให้กำหนดพารามิเตอร์ตามรายละเอียดต่อไปนี้



รูปที่ 5 การกำหนดคอนฟิกสำหรับ Web Deploy

A. กำหนดโฟลเดอร์สำหรับการจัดเก็บไฟล์ OCX เพื่อติดตั้งที่เว็บเซิร์ฟเวอร์

B. กำหนด URL ที่ใช้อ้างถึงเพจที่จะรัน ActiveX Form นี้

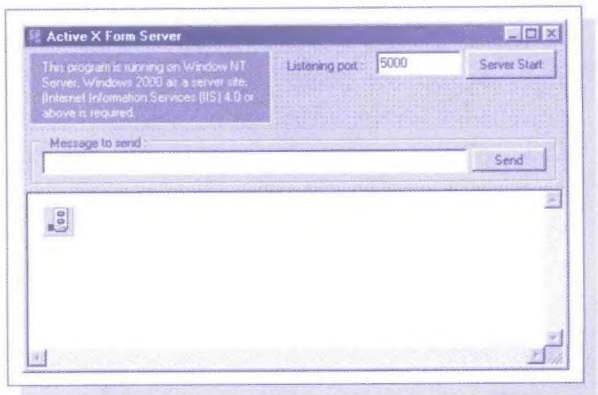
C. กำหนดโฟลเดอร์สำหรับบันทึกไฟล์ HTML เพื่อลงทะเบียน ActiveX Form

D. กรณีที่ต้องการ Compress ไฟล์เพื่อความรวดเร็วในการส่งไฟล์ไปติดตั้งที่เครื่องลูกข่ายให้กาเครื่องหมาย ☑ ที่กรอบนี้ด้วย

E. กรณีที่มีไฟล์เสริมสำหรับร่วมทำงาน และต้องการให้ติดตั้งพร้อมกันกับฟอร์มนี้ ให้กาเครื่องหมาย ☑ ที่กรอบนี้ พร้อมระบุรายการแฟ้มข้อมูลที่ต้องการส่งไปพร้อมกันไว้ในแท็บ Additional Files (G)

F. ในกรณีที่ต้องการให้เดลไฟสร้างหมายเลข Release Version ให้โดยอัตโนมัติ เพื่อใช้ความสามารถในส่วนการติดตั้งอัตโนมัติเมื่อมีการอัพเดตโปรแกรม ให้กาเครื่องหมาย ☑ ที่กรอบนี้ด้วย

เมื่อกำหนด Web Deploy Options เรียบร้อย สามารถสั่งให้เดลไฟสร้างไฟล์ในส่วนของ Web deploy ที่กำหนดได้ โดยเลือกเมนู Project —> Web Deploy เพื่อสร้างส่วนของโปรแกรม



รูปที่ 6 โปรแกรม AxServer (AxServer.Dpr)

AxServer

รูปที่ 6 เป็นโปรแกรม AxServer.Dpr โดยมีส่วนของโปรแกรมแสดงอยู่ในตารางที่ 2 (อยู่ตอนท้ายของบทความ) เป็นส่วนของเซิร์ฟเวอร์แอพพลิเคชัน (Server Application) ที่ทำงานอยู่ที่เว็บเซิร์ฟเวอร์ เพื่อรอรับการติดต่อจากเครื่องลูกข่าย (Clients) ที่รันโปรแกรม ActiveFormProj1 ภายใต้เว็บเพจที่กำหนด

โปรแกรม AxServer มีส่วนของโปรแกรมและกิจกรรมที่สำคัญดังต่อไปนี้

กิจกรรม TForm1.Button1Click(Sender: TObject); กิจกรรมสำหรับเปิดพอร์ตสื่อสารรอรับการติดต่อจากเครื่องลูก

กิจกรรม TForm1.Button3Click(Sender: TObject); กิจกรรมสำหรับดำเนินการส่งข้อมูลให้กับเครื่องลูกที่ติดต่อเข้ามา

กิจกรรม TForm1.ServerSocket1Accept(Sender: TObject; Socket: TCustomWinSocket); กิจกรรมสำหรับกำหนดให้ connectSocket ชี้ไปที่ซ็อกเก็ตของลูกข่ายที่ติดต่อมา

กิจกรรม TForm1.ServerSocket1ClientRead(Sender: TObject; Socket: TCustomWinSocket); เป็นกิจกรรมสำหรับรับข้อมูลที่ส่งมาจากเครื่องลูกข่าย

ทดสอบโปรแกรม

ผู้เขียนขอใช้ระบบวินโดวส์ 2000 สำหรับทดสอบโปรแกรมในส่วนของเซิร์ฟเวอร์ และใช้ระบบวินโดวส์ 98 สำหรับให้บริการในส่วนของเครื่องลูกข่าย ดังนี้

สำหรับเครื่องเซิร์ฟเวอร์

1. รันโปรแกรม Windows 2000 พร้อมตรวจสอบการทำงานของ Internet Information Service เพื่อให้บริการเว็บเซอร์วิสได้

2. สร้างโฟลเดอร์ C:\AXSERVER แล้วทำการคัดลอกโปรแกรม axServer.exe ไปติดตั้ง

3. สร้างโฟลเดอร์ C:\AXFORM\BIN และ C:\AXFORM\HTML

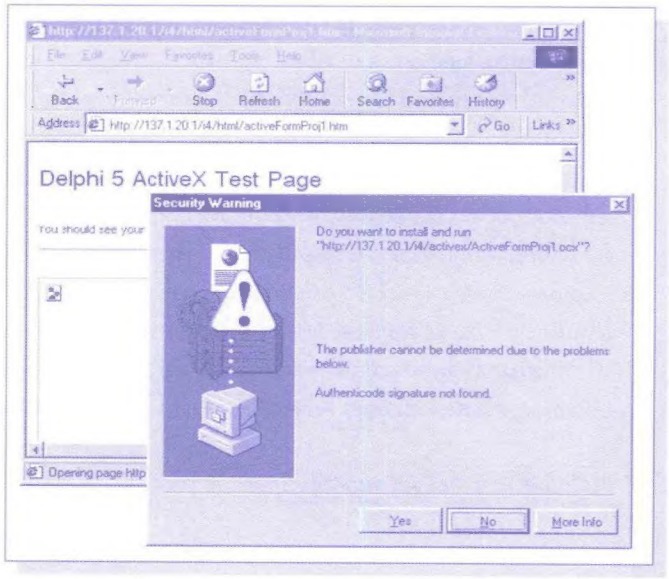
4. คัดลอกทั้งโปรแกรม ActiveFormProj1.ocx และโปรแกรม ActiveFormProj1.htm ไปไว้ในโฟลเดอร์ C:\AXFORM\BIN และ C:\AXFORM\HTML ตามลำดับ

5. สร้าง Virtual Directories ให้กับ Default Website โดยกดปุ่ม [Start] เลือกเมนู Control Panel —> Administrative Tools —> Internet Service Manager เพื่อเปิดหน้าต่าง Internet Information Service

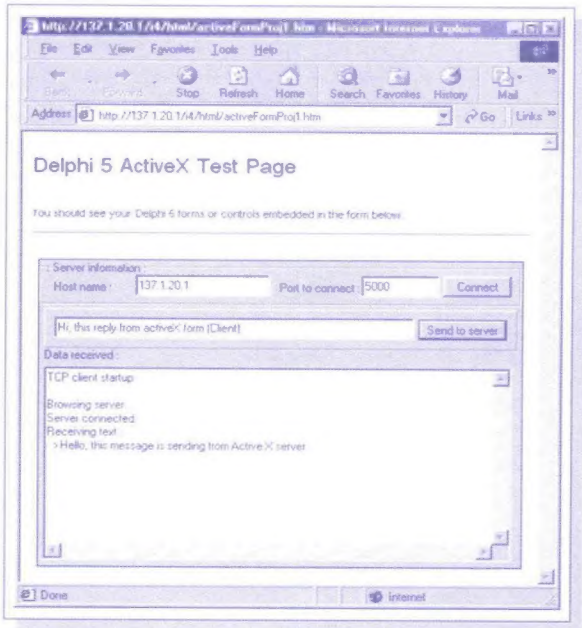
6. เพิ่ม Virtual Directories ชื่อ AXFORM ไว้ภายใต้ Default Website โดยกำหนดลิงค์ให้ชี้ไปที่โฟลเดอร์ C:\AXFORM\BIN

7. แก้ไขไฟล์ ActiveFormProj1.htm ในส่วนของ URL Link ให้ถูกต้องตามความเป็นจริง เช่น (codebase = http://xxx.xxx.xxx.xxx/axform/ActiveFormProj1.ocx#version=1,0,0,0)

8. รันโปรแกรม AxServer และกดปุ่ม [Server Start] เพื่อรอการติดต่อจากเครื่องลูกข่าย



รูปที่ 7 โปรแกรม Internet Explorer จะเตือนการติดตั้ง ActiveX Object ลงในระบบ



รูปที่ 8 หลังจากติดตั้ง ActiveX Form เรียบร้อยและรันทดสอบโปรแกรม

สำหรับเครื่องลูกข่าย

1. รันโปรแกรม Internet Explorer บนระบบวินโดวส์ 98 หรือวินโดวส์ Me

2. สำหรับการใช้งาน ActiveX Form บน Internet Explorer ต้องทำการลดระบบ Security เพื่อที่จะทำให้โปรแกรม Internet Explorer อนุญาตให้ติดตั้ง AxtiveX Form ลงในเครื่องลูกข่ายได้ การลด Security Level สามารถทำได้โดยเปิดหน้าต่าง Internet Options เลือกแท็บ Security เพื่อลดระบบความปลอดภัย

3. เลือก Local Intranet แล้วปรับ Security Level for this zone ให้เป็น medium low หรือต่ำกว่านั้น

4. เลือก Trusted Sites แล้วปรับ Security Level for this zone ให้เป็น medium low หรือต่ำกว่านั้น

5. ทดลอง Browse ไปที่เว็บไซท์ที่กำหนดขึ้น เช่น http://xxx.xxx.xxx.xxx/axform/activeformproj1.htm ถ้าดำเนินการติดตั้งเรียบร้อยถูกต้องตามขั้นตอนทุกประการ โปรแกรม Internet Explorer จะแสดงหน้าต่าง Security Warning ดังรูปที่ 7 เพื่อเตือนการติดตั้ง ActiveX Object ลงในเครื่อง ให้กดปุ่ม [Yes] เพื่อดำเนินการติดตั้ง เมื่อติดตั้งเรียบร้อย โปรแกรม Internet Explorer จะแสดงหน้าต่าง ActiveFormProj1 ในพื้นที่แสดงผล

6. ให้พิมพ์หมายเลข IP Address ของเครื่องเซิร์ฟเวอร์ กดปุ่ม [Connect] โปรแกรมจะทำการติดต่อไปยังเครื่องเซิร์ฟเวอร์ ซึ่งสามารถดำเนินการรับส่งข้อมูลได้ดังรูปที่ 8

ส่งท้าย

จากโปรแกรมที่กล่าวมาแล้วทั้งหมดจะพบว่าเทคโนโลยี ActiveX เป็นเทคโนโลยีที่น่าสนใจ ก่อให้เกิดการเปลี่ยนแปลงแนวความคิดและวิธีการโปรแกรมในอนาคตเป็นอย่างมาก ซึ่งเป็นสิ่งที่กระตุ้นให้นักพัฒนาระบบทุกท่านต้องกระตือรือร้น เพื่อเรียนรู้สิ่งใหม่ ๆ อยู่เสมอ สำหรับบทความในฉบับนี้คงต้องจบลงแต่เพียงเท่านี้ ฉบับหน้าพบกันใหม่ สวัสดีครับ

ตารางที่ 1 ยูนิต ActiveFormImpl1.pas

```
0001 : unit ActiveFormImpl1;
0002 :
0003 : interface
0004 :
0005 : uses
0006 :   Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
0007 :   ActiveX, AxCtrls, ActiveFormProj1_TLB, StdVcl, StdCtrls, ExtCtrls,
0008 :   ScktComp;
0009 :
000A : type
000B :   TActiveFormX = class(TActiveForm, IActiveFormX)
000C :     Memo1: TMemo;
000D :     Label1: TLabel;
000E :     Panel1: TPanel;
000F :     Edit1: TEdit;
0010 :     Button1: TButton;
0011 :     GroupBox2: TGroupBox;
0012 :     Label2: TLabel;
0013 :     Label3: TLabel;
0014 :     Edit2: TEdit;
0015 :     Edit3: TEdit;
0016 :     Button2: TButton;
0017 :     ClientSocket1: TClientSocket;
0018 :     procedure ActiveFormCreate(Sender: TObject);
0019 :     procedure Button2Click(Sender: TObject);
001A :     procedure Button1Click(Sender: TObject);
001B :     procedure ClientSocket1Connect(Sender: TObject;
001C :       Socket: TCustomWinSocket);
001D :     procedure ClientSocket1Error(Sender: TObject; Socket: TCustomWinSocket;
001E :       ErrorEvent: TErrorEvent; var ErrorCode: Integer);
001F :     procedure ClientSocket1Lookup(Sender: TObject;
0020 :       Socket: TCustomWinSocket);
0021 :     procedure ClientSocket1Read(Sender: TObject; Socket: TCustomWinSocket);
0022 :   private
0023 :     { Private declarations }
0024 :     FEvents: IActiveFormXEvents;
0025 :     procedure ActivateEvent(Sender: TObject);
0026 :     procedure ClickEvent(Sender: TObject);
0027 :     procedure CreateEvent(Sender: TObject);
0028 :     procedure DblClickEvent(Sender: TObject);
0029 :     procedure DeactivateEvent(Sender: TObject);
002A :     procedure DestroyEvent(Sender: TObject);
002B :     procedure KeyPressEvent(Sender: TObject; var Key: Char);
002C :     procedure PaintEvent(Sender: TObject);
002D :
002E :     procedure InitVar;
002F :     procedure Echo( content : shortString );
0030 :
0031 :
0032 :   protected
0033 :     { Protected declarations }
0034 :     procedure DefinePropertyPages(DefinePropertyPage: TDefinePropertyPage);
             override;
0035 :     procedure EventSinkChanged(const EventSink: IUnknown); override;
0036 :     function Get_Active: WordBool; safecall;
0037 :     function Get_AutoScroll: WordBool; safecall;
0038 :     function Get_AutoSize: WordBool; safecall;
0039 :     function Get_AxBorderStyle: TxActiveFormBorderStyle; safecall;
003A :     function Get_Caption: WideString; safecall;
003B :     function Get_Color: OLE_COLOR; safecall;
003C :     function Get_Cursor: Smallint; safecall;
003D :     function Get_DoubleBuffered: WordBool; safecall;
003E :     function Get_DropTarget: WordBool; safecall;
003F :     function Get_Enabled: WordBool; safecall;
0040 :     function Get_Font: IFontDisp; safecall;
0041 :     function Get_HelpFile: WideString; safecall;
0042 :     function Get_KeyPreview: WordBool; safecall;
0043 :     function Get_PixelsPerInch: Integer; safecall;
0044 :     function Get_PrintScale: TxPrintScale; safecall;
0045 :     function Get_Scaled: WordBool; safecall;
0046 :     function Get_Visible: WordBool; safecall;
0047 :     function Get_VisibleDockClientCount: Integer; safecall;
0048 :     procedure _Set_Font(const Value: IFontDisp); safecall;
0049 :     procedure Set_AutoScroll(Value: WordBool); safecall;
004A :     procedure Set_AutoSize(Value: WordBool); safecall;
004B :     procedure Set_AxBorderStyle(Value: TxActiveFormBorderStyle); safecall;
004C :     procedure Set_Caption(const Value: WideString); safecall;
004D :     procedure Set_Color(Value: OLE_COLOR); safecall;
004E :     procedure Set_Cursor(Value: Smallint); safecall;
004F :     procedure Set_DoubleBuffered(Value: WordBool); safecall;
0050 :     procedure Set_DropTarget(Value: WordBool); safecall;
0051 :     procedure Set_Enabled(Value: WordBool); safecall;
0052 :     procedure Set_Font(var Value: IFontDisp); safecall;
0053 :     procedure Set_HelpFile(const Value: WideString); safecall;
0054 :     procedure Set_KeyPreview(Value: WordBool); safecall;
0055 :     procedure Set_PixelsPerInch(Value: Integer); safecall;
0056 :     procedure Set_PrintScale(Value: TxPrintScale); safecall;
0057 :     procedure Set_Scaled(Value: WordBool); safecall;
0058 :     procedure Set_Visible(Value: WordBool); safecall;
0059 :   public
005A :     { Public declarations }
005B :     procedure Initialize; override;
005C :   end;
005D :
005E : implementation
005F :
0060 : uses ComObj, ComServ;
0061 :
0062 : {$R *.DFM}
0063 :
0064 : { TActiveFormX }
0065 :
0066 : procedure TActiveFormX.DefinePropertyPages(DefinePropertyPage:
TDefinePropertyPage);
0067 : begin
0068 :   { Define property pages here.  Property pages are defined by calling
0069 :     DefinePropertyPage with the class id of the page.  For example,
006A :       DefinePropertyPage(Class_ActiveFormXPage); }
006B : end;
```

ตารางที่ 1 (ต่อ)

```
006C :
006D : procedure TActiveFormX.EventSinkChanged(const EventSink: IUnknown);
006E : begin
006F :   FEvents := EventSink as IActiveFormXEvents;
0070 : end;
0071 :
0072 : procedure TActiveFormX.Initialize;
0073 : begin
0074 :   inherited Initialize;
0075 :   OnActivate := ActivateEvent;
0076 :   OnClick := ClickEvent;
0077 :   OnCreate := CreateEvent;
0078 :   OnDblClick := DblClickEvent;
0079 :   OnDeactivate := DeactivateEvent;
007A :   OnDestroy := DestroyEvent;
007B :   OnKeyPress := KeyPressEvent;
007C :   OnPaint := PaintEvent;
007D : end;
007E :
007F : function TActiveFormX.Get_Active: WordBool;
0080 : begin
0081 :   Result := Active;
0082 : end;
0083 :
0084 : function TActiveFormX.Get_AutoScroll: WordBool;
0085 : begin
0086 :   Result := AutoScroll;
0087 : end;
0088 :
0089 : function TActiveFormX.Get_AutoSize: WordBool;
008A : begin
008B :   Result := AutoSize;
008C : end;
008D :
008E : function TActiveFormX.Get_AxBorderStyle: TxActiveFormBorderStyle;
008F : begin
0090 :   Result := Ord(AxBorderStyle);
0091 : end;
0092 :
0093 : function TActiveFormX.Get_Caption: WideString;
0094 : begin
0095 :   Result := WideString(Caption);
0096 : end;
0097 :
0098 : function TActiveFormX.Get_Color: OLE_COLOR;
0099 : begin
009A :   Result := OLE_COLOR(Color);
009B : end;
009C :
009D : function TActiveFormX.Get_Cursor: Smallint;
009E : begin
009F :   Result := Smallint(Cursor);
00A0 : end;
00A1 :
00A2 : function TActiveFormX.Get_DoubleBuffered: WordBool;
00A3 : begin
00A4 :   Result := DoubleBuffered;
00A5 : end;
00A6 :
00A7 : function TActiveFormX.Get_DropTarget: WordBool;
00A8 : begin
00A9 :   Result := DropTarget;
00AA : end;
00AB :
00AC : function TActiveFormX.Get_Enabled: WordBool;
00AD : begin
00AE :   Result := Enabled;
00AF : end;
00B0 :
00B1 : function TActiveFormX.Get_Font: IFontDisp;
00B2 : begin
00B3 :   GetOleFont(Font, Result);
00B4 : end;
00B5 :
00B6 : function TActiveFormX.Get_HelpFile: WideString;
00B7 : begin
00B8 :   Result := WideString(HelpFile);
00B9 : end;
00BA :
00BB : function TActiveFormX.Get_KeyPreview: WordBool;
00BC : begin
00BD :   Result := KeyPreview;
00BE : end;
00BF :
00C0 : function TActiveFormX.Get_PixelsPerInch: Integer;
00C1 : begin
00C2 :   Result := PixelsPerInch;
00C3 : end;
00C4 :
00C5 : function TActiveFormX.Get_PrintScale: TxPrintScale;
00C6 : begin
00C7 :   Result := Ord(PrintScale);
00C8 : end;
00C9 :
00CA : function TActiveFormX.Get_Scaled: WordBool;
00CB : begin
00CC :   Result := Scaled;
00CD : end;
00CE :
00CF : function TActiveFormX.Get_Visible: WordBool;
00D0 : begin
00D1 :   Result := Visible;
00D2 : end;
00D3 :
00D4 : function TActiveFormX.Get_VisibleDockClientCount: Integer;
00D5 : begin
00D6 :   Result := VisibleDockClientCount;
00D7 : end;
00D8 :
00D9 : procedure TActiveFormX._Set_Font(const Value: IFontDisp);
00DA : begin
00DB :   SetOleFont(Font, Value);
00DC : end;
00DD :
00DE : procedure TActiveFormX.ActivateEvent(Sender: TObject);
00DF : begin
00E0 :   if FEvents <> nil then FEvents.OnActivate;
00E1 : end;
00E2 :
00E3 : procedure TActiveFormX.ClickEvent(Sender: TObject);
00E4 : begin
00E5 :   if FEvents <> nil then FEvents.OnClick;
00E6 : end;
00E7 :
00E8 : procedure TActiveFormX.CreateEvent(Sender: TObject);
00E9 : begin
00EA :   if FEvents <> nil then FEvents.OnCreate;
00EB : end;
00EC :
00ED : procedure TActiveFormX.DblClickEvent(Sender: TObject);
00EE : begin
00EF :   if FEvents <> nil then FEvents.OnDblClick;
00F0 : end;
00F1 :
00F2 : procedure TActiveFormX.DeactivateEvent(Sender: TObject);
00F3 : begin
00F4 :   if FEvents <> nil then FEvents.OnDeactivate;
00F5 : end;
00F6 :
00F7 : procedure TActiveFormX.DestroyEvent(Sender: TObject);
00F8 : begin
00F9 :   if FEvents <> nil then FEvents.OnDestroy;
00FA : end;
00FB :
00FC : procedure TActiveFormX.KeyPressEvent(Sender: TObject; var Key: Char);
00FD : var
00FE :   TempKey: Smallint;
00FF : begin
0100 :   TempKey := Smallint(Key);
0101 :   if FEvents <> nil then FEvents.OnKeyPress(TempKey);
0102 :   Key := Char(TempKey);
0103 : end;
```

ตารางที่ 1 (ต่อ)

```
0104 :
0105 : procedure TActiveFormX.PaintEvent(Sender: TObject);
0106 : begin
0107 :   if FEvents <> nil then FEvents.OnPaint;
0108 : end;
0109 :
010A : procedure TActiveFormX.Set_AutoScroll(Value: WordBool);
010B : begin
010C :   AutoScroll := Value;
010D : end;
010E :
010F : procedure TActiveFormX.Set_AutoSize(Value: WordBool);
0110 : begin
0111 :   AutoSize := Value;
0112 : end;
0113 :
0114 : procedure TActiveFormX.Set_AxBorderStyle(Value:
TxActiveFormBorderStyle);
0115 : begin
0116 :   AxBorderStyle := TActiveFormBorderStyle(Value);
0117 : end;
0118 :
0119 : procedure TActiveFormX.Set_Caption(const Value: WideString);
011A : begin
011B :   Caption := TCaption(Value);
011C : end;
011D :
011E : procedure TActiveFormX.Set_Color(Value: OLE_COLOR);
011F : begin
0120 :   Color := TColor(Value);
0121 : end;
0122 :
0123 : procedure TActiveFormX.Set_Cursor(Value: Smallint);
0124 : begin
0125 :   Cursor := TCursor(Value);
0126 : end;
0127 :
0128 : procedure TActiveFormX.Set_DoubleBuffered(Value: WordBool);
0129 : begin
012A :   DoubleBuffered := Value;
012B : end;
012C :
012D : procedure TActiveFormX.Set_DropTarget(Value: WordBool);
012E : begin
012F :   DropTarget := Value;
0130 : end;
0131 :
0132 : procedure TActiveFormX.Set_Enabled(Value: WordBool);
0133 : begin
0134 :   Enabled := Value;
0135 : end;
0136 :
0137 : procedure TActiveFormX.Set_Font(var Value: IFontDisp);
0138 : begin
0139 :   SetOleFont(Font, Value);
013A : end;
013B :
013C : procedure TActiveFormX.Set_HelpFile(const Value: WideString);
013D : begin
013E :   HelpFile := String(Value);
013F : end;
0140 :
0141 : procedure TActiveFormX.Set_KeyPreview(Value: WordBool);
0142 : begin
0143 :   KeyPreview := Value;
0144 : end;
0145 :
0146 : procedure TActiveFormX.Set_PixelsPerInch(Value: Integer);
0147 : begin
0148 :   PixelsPerInch := Value;
0149 : end;
014A :
014B : procedure TActiveFormX.Set_PrintScale(Value: TxPrintScale);
014C : begin
014D :   PrintScale := TPrintScale(Value);
014E : end;
014F :
0150 : procedure TActiveFormX.Set_Scaled(Value: WordBool);
0151 : begin
0152 :   Scaled := Value;
0153 : end;
0154 :
0155 : procedure TActiveFormX.Set_Visible(Value: WordBool);
0156 : begin
0157 :   Visible := Value;
0158 : end;
0159 :
015A : procedure TActiveFormX.ActiveFormCreate(Sender: TObject);
015B : begin
015C :   InitVar;
015D : end;
015E :
015F : //-------------------------------------------------------------
0160 :
0161 : procedure TActiveFormX.InitVar;
0162 : begin
0163 :
0164 : end;
0165 :
0166 : procedure TActiveFormX.Echo(content: shortString);
0167 : begin
0168 :   memo1.lines.add( content );
0169 : end;
016A :
016B : //-------------------------------------------------------------
016C :
016D : procedure TActiveFormX.Button2Click(Sender: TObject);
016E : var hostName : shortString; portNo : integer;
016F : begin
0170 :   hostName:=Edit2.text;
0171 :   if ( hostName='' ) then
0172 :   begin Echo( 'Host name not specified' ); exit; end;
0173 :   Echo( 'TCP client startup'+#13#10 );
0174 :   portNo:=StrToInt( Edit3.text );
0175 :   ClientSocket1.Host:=hostName;
0176 :   ClientSocket1.Port:=portNo;
0177 :   ClientSocket1.Open;
0178 : end;
0179 :
017A : procedure TActiveFormX.Button1Click(Sender: TObject);
017B : var s : shortString; p : pchar;
017C : begin
017D :   s:=Edit1.text; p:=@s[1];
017E :   clientSocket1.Socket.SendBuf( p^, length( s ) );
017F : end;
0180 :
0181 : procedure TActiveFormX.ClientSocket1Connect(Sender: TObject;
0182 :   Socket: TCustomWinSocket);
0183 : begin
0184 :   Echo( 'Server connected.' );
0185 : end;
0186 :
0187 : procedure TActiveFormX.ClientSocket1Error(Sender: TObject;
0188 :   Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
0189 :   var ErrorCode: Integer);
018A : begin
018B :   Echo( ':ERROR' );
018C :   Echo( '  Error code = '+IntToStr( errorCode ) );
018D : end;
018E :
018F : procedure TActiveFormX.ClientSocket1Lookup(Sender: TObject;
0190 :   Socket: TCustomWinSocket);
0191 : begin
0192 :   Echo( 'Browsing server..' );
0193 : end;
0194 :
0195 : procedure TActiveFormX.ClientSocket1Read(Sender: TObject;
0196 :   Socket: TCustomWinSocket);
0197 : var s : shortString; p : pchar; size : integer;
0198 : begin
0199 :   fillchar( s, sizeof( shortString ), 0 ); p:=@s[1];
019A :   size:=socket.ReceiveBuf( p^, sizeof( shortString ) );
019B :   s[0]:=chr(size);
```

ตารางที่ 1 (ต่อ)

```
019C :   Echo( 'Receiving text : ' );
019D :   Echo( '  >'+s );
019E : end;
019F :
01A0 : initialization
01A1 :   TActiveFormFactory.Create(
01A2 :     ComServer,
01A3 :     TActiveFormControl,
01A4 :     TActiveFormX,
01A5 :     Class_ActiveFormX,
01A6 :     1,
01A7 :     '',
01A8 :     OLEMISC_SIMPLEFRAME or OLEMISC_ACTSLIKELABEL,
01A9 :     tmApartment);
01AA : end.
```

ตารางที่ 2 ยูนิต uaxSrvr.pas

```
0001 : unit uaxSrvr;
0002 :
0003 : interface
0004 :
0005 : uses
0006 :   Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
0007 :   StdCtrls, ScktComp, ExtCtrls;
0008 :
0009 : type
000A :   TForm1 = class(TForm)
000B :     Memo1: TMemo;
000C :     Button1: TButton;
000D :     ServerSocket1: TServerSocket;
000E :     Edit1: TEdit;
000F :     Label1: TLabel;
0010 :     GroupBox3: TGroupBox;
0011 :     Edit2: TEdit;
0012 :     Button3: TButton;
0013 :     Panel1: TPanel;
0014 :     StaticText1: TStaticText;
0015 :     procedure Button1Click(Sender: TObject);
0016 :     procedure Button3Click(Sender: TObject);
0017 :     procedure FormCreate(Sender: TObject);
0018 :     procedure FormCloseQuery(Sender: TObject; var CanClose: Boolean);
0019 :     procedure ServerSocket1Accept(Sender: TObject;
001A :       Socket: TCustomWinSocket);
001B :     procedure ServerSocket1ClientError(Sender: TObject;
001C :       Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
001D :       var ErrorCode: Integer);
001E :     procedure ServerSocket1ClientRead(Sender: TObject;
001F :       Socket: TCustomWinSocket);
0020 :     procedure ServerSocket1Listen(Sender: TObject;
0021 :       Socket: TCustomWinSocket);
0022 :   private
0023 :     { Private declarations }
0024 :       connectSocket : TCustomWinSocket;
0025 :     procedure InitVar;
0026 :     procedure ReleaseVar;
0027 :     procedure Echo( content : shortString );
0028 :   public
0029 :     { Public declarations }
002A :   end;
002B :
002C : var
002D :   Form1: TForm1;
002E :
002F : implementation
0030 :
0031 : {$R *.DFM}
0032 :
0033 : //-------------------------------------------------------------------
0034 :
0035 : procedure TForm1.InitVar;
0036 : begin
0037 :   connectSocket:=nil;
0038 : end;
0039 :
003A : procedure TForm1.ReleaseVar;
003B : begin
003C :
003D : end;
003E :
003F : procedure TForm1.Echo(content: shortString);
0040 : begin
0041 :   memo1.lines.add( content );
0042 : end;
0043 :
0044 : //-------------------------------------------------------------------
0045 :
0046 : procedure TForm1.Button1Click(Sender: TObject);
0047 : var portNo : integer;
0048 : begin
0049 :   portNo:=StrToIntDef( edit1.text,5000 );
004A :   ServerSocket1.port:=portNo;
004B :   ServerSocket1.Open;
004C : end;
004D :
004E : procedure TForm1.Button3Click(Sender: TObject);
004F : var s : shortString; p : pchar;
0050 : begin
0051 :   if ( connectSocket=nil ) then exit;
0052 :   fillchar( s, sizeof( shortString ), 0 );
0053 :   s:=Edit2.text; p:=@s[1];
0054 :   connectSocket.SendBuf( p^, length( s ) );
0055 : end;
0056 :
0057 : procedure TForm1.FormCreate(Sender: TObject);
0058 : begin
0059 :   InitVar;
005A : end;
005B :
005C : procedure TForm1.FormCloseQuery(Sender: TObject; var CanClose: Boolean);
005D : begin
005E :   ReleaseVar;
005F : end;
0060 :
0061 : procedure TForm1.ServerSocket1Accept(Sender: TObject;
0062 :   Socket: TCustomWinSocket);
0063 : begin
0064 :   Echo( 'Client '+socket.RemoteAddress+' is connected.' );
0065 :   connectSocket:=Socket;
0066 : end;
0067 :
0068 : procedure TForm1.ServerSocket1ClientError(Sender: TObject;
0069 :   Socket: TCustomWinSocket; ErrorEvent: TErrorEvent;
006A :   var ErrorCode: Integer);
006B : begin
006C :   Echo( ':ERROR' );
006D :   Echo( '  Error code = '+IntToStr( errorCode ) );
006E : end;
006F :
0070 : procedure TForm1.ServerSocket1ClientRead(Sender: TObject;
0071 :   Socket: TCustomWinSocket);
0072 : var s : shortString; p : pchar; size : integer;
0073 : begin
0074 :   fillchar( s, sizeof( shortString ), 0 ); p:=@s[1];
0075 :   size:=socket.ReceiveBuf( p^, sizeof( shortString ) );
0076 :   s[0]:=chr(size);
0077 :   Echo( 'Receiving message : ' );
0078 :   Echo( '  >'+s );
0079 : end;
007A :
007B : procedure TForm1.ServerSocket1Listen(Sender: TObject;
007C :   Socket: TCustomWinSocket);
007D : begin
007E :   Echo( 'Socket open, waiting for request on' );
007F :   Echo( '  IP   : '+Socket.LocalAddress );
0080 :   Echo( '  Port : '+IntToStr( Socket.LocalPort ) );
0081 :   Echo( '' ); Echo( 'Waiting for client connect..' );
0082 : end;
0083 :
0084 : end.
0085 :
```